Journal of Computer and Knowledge Engineering, Vol. 6, No. 1, 2023. (81-100)
DOI: 10.22067/cke.2023.79741.1071

A Novel Correlation-Based Feature Selection Approach using Manta Ray Foraging Optimization*

Research Article

Najme Mansouri[1] Mohammad Ansari Shiri[2]

Abstract: Recent advances in science, engineering, and technology have created massive datasets. As a result, machine learning and data mining techniques cannot perform well on these huge datasets, because they contain redundant, noisy, and irrelevant features. The purpose of feature selection is reducing the dimensionality of datasets by selecting the most relevant attributes while simultaneously increasing classification accuracy. The application of meta-heuristic optimization techniques has become increasingly popular for feature selection in recent years due to their ability to overcome the limitations of traditional optimization methods. This paper presents a binary version of the Manta Ray Foraging Optimizer (MRFO), an alternative optimization algorithm. Besides reducing costs and reducing calculation time, we also incorporated Spearman's correlation coefficient into the proposed method, which we called Correlation Based Binary Manta Ray Foraging (CBBMRF). It eliminates highly positive correlation features at the beginning of the calculation, avoiding additional calculations and leading to faster subset selection. A comparison is made between the presented algorithms and five state-of-the-art meta-heuristics using 10 standard UCI datasets. As a result, the proposed algorithms demonstrate superior performance when solving feature selection problems.

Keywords: Feature Selection, Optimization, Correlation, Accuracy.

1. Introduction

Learning algorithms are typically underperforming when faced with noisy, redundant, and meaningless classification datasets. To reduce the dimensionality of the datasets, feature selection is a preprocessing step [1]. As a tool for reducing the dimensionality of data, improving prediction accuracy, and understanding it, feature selection is often used in machine learning applications such as clustering, classification, regression, and computer vision [2]. The selection of relevant features can be useful in supervised learning as well, as they maximize prediction accuracy by optimizing certain functions. Several feature selection techniques have been developed and used for the optimization of the predictive model [3]. Often, IoT applications generate big data from sensor nodes, and that data must be analyzed. Sensor nodes' performance is hampered by such factors as energy consumption, storage, processing power, and distance from communication networks. As a result of Feature Selection (FS), big data generated from IoT can be reduced in dimension and the unwanted data can be ignored, which simplifies the task of processing [4]. There are many problems associated with feature selection. One of the most significant and common problems is the curse of dimensionality, which results in problems that reduce accuracy and slow learning speed when the attributes or numbers of the features exceed the samples. Consequently, datasets should be summarized in order to reduce noise and redundancy while identifying smaller or narrower attributes. This process, called dimensionality reduction, leads to better classification performance and enhanced discrimination power [5].

Feature subsets that are near optimal have been identified using metaheuristic algorithms in recent decades. Metaheuristic search methods show better performance when compared to exact search methods due to the ability of searching the entire search space [6]. A metaheuristic algorithm consists of two important features: Exploration and exploitation. Every time that a new solution is sought, exploration involves searching the entire solution space without making any assumptions about its local optimum location. In the exploitation process, a better solution is found in the neighborhood of the solution obtained, which speeds up convergence. Exploration and exploitation should be balanced in a good meta-heuristic algorithm.

The presence of many metaheuristic and hybrid metaheuristic FS strategies strongly suggests that another hybrid meta-heuristic FS algorithm is needed. According to the No Free Lunch theorem, no single optimization algorithm can solve all optimization problems. This research focuses primarily on giving the algorithm some new facet that combines exploration and exploitation to achieve a superior trade-off for each new algorithm following any regular or natural phenomenon. Ultimately, it reaches the global optimality by diverging from the local optima. However, achieving these objectives isn't as simple as it sounds, especially when one has to propose an algorithm that can be used in several domains. In order to keep the research area alive, researchers formulate better methods over time.

It is almost impossible to find an optimal value for each dimension at the same time when considering a multimodal optimization problem. In order to solve these problems within reasonable timeframes, researchers use metaheuristic strategies. Multiple optimal subsets are possible with FS, i.e., subsets with the same precision and dimension as the original subset. It is possible to have more than one optimal subset because FS is an optimization problem. It would also be extremely difficult to find a feature subset whose storage space, running time, and performance of the machine learning algorithm were optimal.

To meet these prerequisites, research is still ongoing [7].

---

* Manuscript received: 2022 November 23, Revised, 2023 April 4, Accepted, 2023June 21.
1. Corresponding author. PhD in Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.
Email: najme.mansouri@gmail.com.
2. MSc. in Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.

This has inspired us to propose a meta-heuristic FS method based on Manta Ray Foraging Optimizer (MRFO) [8] algorithm. This algorithm is based on the intelligent behavior of manta rays. Three unique foraging strategies for manta rays are mimicked by this work: Chain foraging, cyclone foraging, and somersault foraging. To improve the performance of the proposed method, we added the correlation coefficient [9] approach to the binary type of MRFO algorithm (CBBMRF). MRFO is used for the first time in FS. This paper contributes the following novelties:

1. The most recent meta-heuristic MRFO is used in the development of a new FS method known as BMRF algorithm;
2. A binary version of MRFO is presented;
3. The correlation coefficient method is added to the proposed algorithm (BMRF) to improve its performance;
4. A FS approach is evaluated using ten standard UCI datasets;
5. The proposed FS approach and 5 meta-heuristic-based FS methods are compared;
6. The evaluation of the proposed feature selection is done by two classifiers (i.e., Random Forest (RF) and K-Nearest Neighbors (KNN) classifier).

The remainder of the paper is as follows. The second section describes feature selection, meta-heuristic algorithms, and learning models. Section 3 discusses related works on feature selection using evolutionary algorithms. An overview of the proposed schedule of work is provided in Section 4. Section 5 presents the experimental setup and results and Section 6 deals with the conclusion.

2. Background

2.1. Feature selection

A discussion of feature selection approaches and strategies is presented in this section. Figure 1 illustrates how features are selected. There are three broad categories of traditional feature selection approaches for machine learning [10]: Filter method, wrapper method, and embedded method.

Filter-based feature selection: The importance of the feature for addition in the subset of features is assessed based on the important characteristics of the data. Rank-based and subset-based evaluations fall into two categories. In the rank-based category, each attribute is ranked individually without considering the interrelationships between them. Redundant features cannot be identified using this method. Using multivariate statistical techniques, the entire feature subset is evaluated using the subset evaluation-based category. Among the advantages of multivariate statistical techniques are the consideration of feature dependency, the absence of a classifier, and a more efficient computational approach than wrapper techniques. When compared to the univariate ranking method, the multivariate ranking method is slower and less stable. In terms of accuracy and stability, the Joint Mutual Information and Maximum Minimum Nonlinear Approach filter techniques produce the best results [11].

Wrapper-based feature selection: Through wrapper approach, prominent features are evaluated by a classifier that has been trained. Wrapper model evaluates a subset of features in their search processes for the purpose of selecting the most accurate feature set. Iterative search processes are used in most wrapper methods, where each iteration of the learning model guides the population of solutions towards the optimal solution. Despite this, wrapper approaches often incur high computational costs and lose generality due to the involvement of learning models in search processes. It is fast enough to use filter methods and their results are not affected by specific classifiers [12].

Embedded-based feature selection: Selecting features and classifying them are combined during the modeling algorithm's execution in order to maximize the efficiency of the feature selection process. Different decision tree algorithms (e.g., CART or random forest) as well as least square regression and support vector machines (SVM) are most commonly used. In their design, filters and wrappers are combined to provide the best of both worlds. As a result, the feature space is first reduced with a filter and then selected with a wrapper [13].

2.2. Meta-heuristic algorithm

Metaheuristics are routinely referred to as "shortest way" of solving problems, because they determine the best and most feasible solution from all possible alternatives. By analyzing the predicted best solutions, it also estimates each potential solution's ability by performing a series of operations to discover better alternatives.

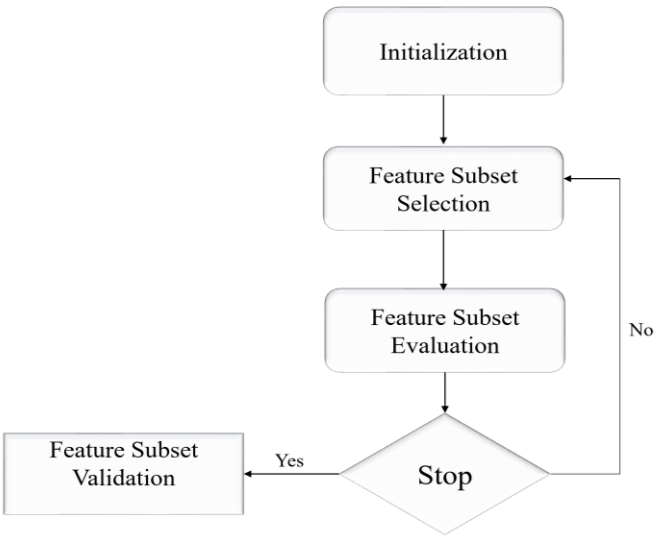


Figure 1. Feature selection process [9]

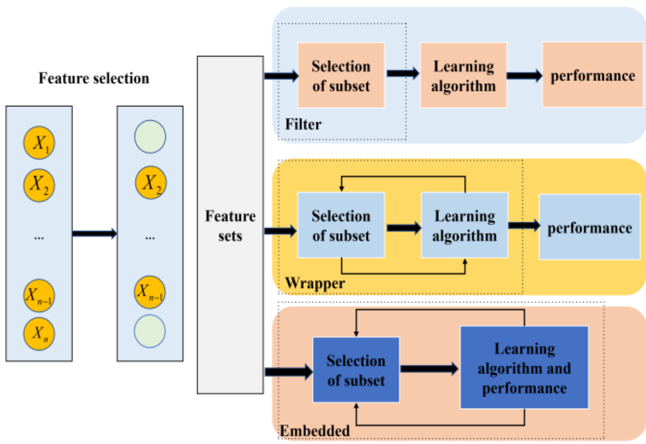


Figure 2. Filter, wrapper, and embedded process [14]

It is not uncommon for metaheuristic algorithms to be combined with optimization techniques in order to find proper solutions from a wide range of viable solutions with minimum computational effort. Figure 2 illustrates how metaheuristic algorithms select features by performing the following steps. Metaheuristic algorithms are categorized into five classes in terms of their availability:

Bio-inspired algorithms**:** A metaheuristic optimization algorithm is based on the biological behavior of any living organism that has the ability to predict the optimal solution to a given problem regarding some constraints on the search space. Bioinspired algorithms simulate the behavior of biological creatures while attacking for food and mates, and are based on the behavior of these creatures. It introduces alternatives to complex problems by exploring their logic and thinking abilities in order to reduce the effort taken by biological creatures at different times. Most commonly used bio-inspired algorithms are Krill Herd Algorithm (KHA), Artificial Immune System (AIS), Bacterial Foraging Optimization (BFO), and Dendritic Cell Algorithm (DCA).

Nature-inspired algorithms**:** The goal of an optimization algorithm is to find the optimal solution to a problem with a limited search space using methods and strategies. Natural-inspired algorithms mimic the behavior of mammals and birds while attacking for food and mates, and are referred to as nature-inspired algorithms. Cuckoo Search (CS), Invasive Weed Optimization Algorithm (IWO), and Firefly Algorithm (FA) are examples of nature-inspired algorithms.

Physics-based algorithms**:** In physics-inspired algorithms, feasible solutions are found both globally and locally from a set of solutions with given constraints. It relies on natural phenomena occurring in our environment under certain conditions, which involves particles and atoms. Memetic Algorithms (MA), Gravitational Search Algorithms (GSA), and Harmony Search algorithms are some of the most well-known physics-inspired algorithms.

Evolutionary algorithms: In addition to metaheuristics, evolutionary algorithms solve NP problems easily that take longer duration of time to solve, i.e., they solve NP problems in polynomial time. Biological evolution and natural selection are also incorporated into the process, with four major steps, starting with initialization, selection, genetic operators, and concluding with termination. Differential Evolution (DE), Genetic Programming (GP), and Genetic Algorithm (GA) are the most commonly used evolutionary algorithms.

Swarm-based algorithms**:** A swarm-based algorithm is an artificial intelligence technology combining the natural and artificial behaviors of a group of individuals in a solution set controlled by themselves. A flock of birds, a group of animals, a colony of ants, or a school of fish can all control each other without any centralized authority. These algorithms are commonly applied in artificial bee colonies, particle swarm optimization, ant colonies and fish swarm algorithms [15].

It is essential to balance exploration (diversification) and exploitation (intensification) activities in metaheuristic algorithms in order for them to perform optimally, predict correctly, and converge rapidly. It is not yet clear what the answer to this question will be. By using fitness landscape analysis and information landscape approaches, we can find a better balance between these activities. There should not be a 50% split between exploration and exploitation in the total optimization process. Dynamic algorithms should be good at resolving issues like this. As a result, metaheuristics typically perform better than other algorithms because they can adapt to these phases. There should be an even distribution of visits to unexplored areas, and the search should not become stymied by local optimum points. During exploration, local optima are usually discarded, while during exploitation, neighboring alternatives are sought. The solution changes in a major way after this technique is applied. Searching for promising regions can be improved by using exploitation operators. It is used to change the feature values if they are one, and zero if they are zero [16]. As Figure 3 shows, metaheuristic algorithms perform the main steps for feature selection optimization.

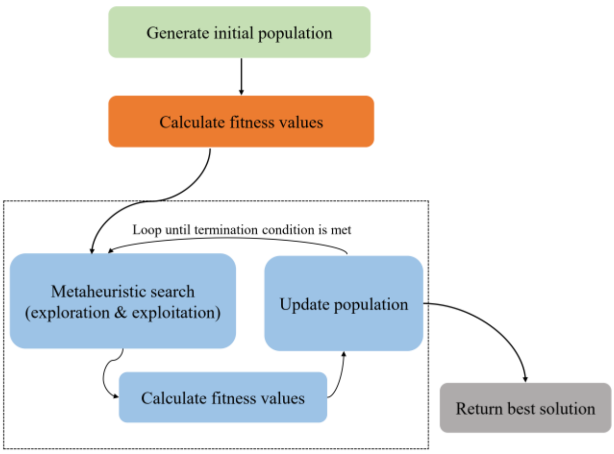


Figure 3. The main steps for the optimization of feature selection using metaheuristic algorithms [16]

Feature selection and meta-heuristic algorithms are useful tools for reviewing, comparing, and evaluating meta-heuristic approaches. Kurman and Kisan [17] reviewed meta-heuristic approaches in depth. Globally, cervical cancer affects more than 80 million women, most of whom live in low-income countries such as India. According to the literature, cervical cancer can be detected early and accurately diagnosed to increase survival rates. When it is in its early stages, this disease does not exhibit any symptoms. With the use of machine learning and deep learning techniques, cervical cancer can be rapidly, accurately and regularly classified and diagnosed, allowing the patient's health to be monitored on an ongoing basis. Feature selection has been solved independently by meta-heuristic algorithms for decades. They provide an alternative solution to the global problem. A number of studies have described the use of feature selection techniques to detect cervical cancer, but no survey has been conducted. They summarize the methods for selecting features in cervical cancer data so that a research gap can be identified, which guides researchers in their future endeavors. Various classifications of techniques are also provided, including those based on nature or not inspired by nature and those based on pathways or populations. There is also a discussion of classical feature

selection techniques for cervical cancer classification as well as meta-heuristic algorithms that can be used to select features.

Yadav et al. [18] presented feature selection and classification techniques that enhance the performance and security of WSNs for IoT applications. During wireless sensor network (WSN) research, energy consumption, secure connectivity, and performance analysis play crucial roles. Moreover, IoT applications have resulted in complex networks due to increased usage. For addressing threats and security issues in complex WSNs, a fast correlation-based feature selection method with XG-Boost is proposed using the NSL-KDD benchmark dataset. Cluster-based WSNs can then be classified using the best features selected before. The presented research develops a robust intrusion detection system for WSNs and their IoT applications using five popular machine learning classifiers: Decision trees, random forests, Nave Bayes, additive trees, and XG boosts.

Ebrahimi and Hemmati [19] used a multi-objective gravity search algorithm to design a complementary voltage-controlled oscillator. Voltage-controlled oscillators (VCO) have been developed rapidly in the industrial and academic communities in recent decades. Therefore, an optimal design for a complementary cross-coupled LC-VCO is achieved using a new multi-objective optimization method. With an oscillation frequency of 2.5 GHz and a supply voltage of 1.5 volts, the design objective is to minimize phase noise and power consumption. A cross-coupled configuration popular with semiconductor manufacturers, the complementary LC-VCO, is described sufficiently in this article. Moreover, the verification theorems for the proposed method indicate that it can control the algorithm's exploration and exploitation capabilities. MOGSA outperforms other multi-objective methods because it is improved.

2.3. Manta *Ray Foraging Optimizer* (MRFO)

MRFO is an innovative, bio-inspired optimization technique aimed at providing an alternate way of optimizing real-world engineering problems [8]. A manta ray's intelligent behavior inspired this algorithm. MRFO paradigm adopts three unique foraging strategies exhibited by manta rays, including chain foraging, cyclone foraging, and somersault foraging. The model is designed to solve different optimization problems efficiently. As manta rays lack sharp teeth, they feed mostly on plankton, which is composed of microscopic animals. The horn-shaped cephalic lobes on their heads funnel water and prey into their mouths when they are foraging. In the next step, modified gill rakers filter the prey from the water.

2.4. Mathematical model of MRFO

MRFO draws inspiration from chains, cyclones, and somersaults of foraging behaviors. Detailed descriptions of the mathematical models can be found below [8].

Chain foraging: Manta rays can swim to plankton based on their observation of their position in MRFO. Positions with a higher concentration of plankton are better than those with a lower concentration. It is unclear which solution is the best, but MRFO believes it is the plankton with high concentrations that manta rays are looking for and wanting to eat. During foraging, manta rays form a chain from head to tail. In addition to moving towards the food, the individuals around it move as well. The best solution found so far, along with the solution in front, is presented to each individual in every iteration. Chain foraging can be represented mathematically as follows:

$$x_i^d(t+1) = \begin{cases} x_i^d(t)+r.(x_{best}^d(t)-x_i^d(t))+\alpha.(x_{best}^d-x_i^d(t)), i=1 \\ x_i^d(t)+r.(x_{i-1}^d(t)-x_i^d(t))+\alpha.(x_{best}^d(t)-x_i^d(i)), i=2,\dots,N \end{cases} \quad (1)$$

$$\alpha = 2.r.\sqrt{|\,log(r)|} \quad (2)$$

An individual's position at time t is termed $x_i^d(t)$, while r represents a random vector within the range of 0 to 1. α indicates the weight coefficient and $x_{best}^d(t)$ indicates the concentration of plankton. Its position is updated based on the position $x_{i-1}(t)$ of the *(i-1)-th* current individual and the food's position $x_{best}(t)$.

Cyclone foraging: Upon detecting plankton in deep waters, manta rays swim towards it in a spiral pattern, forming a long foraging chain. Similarly, WOA uses spiral foraging strategies. In swarms of manta rays, however, each manta ray swims towards its predecessor in addition to spirally moving towards the food. A swarm of manta rays forages in a spiral fashion. Individuals do not only follow their leaders, but also follow a spiral path towards food. In terms of a mathematical equation, manta rays move spiral-shaped in 2-D space as follows:

$$\begin{cases} X_i(t+1)=X_{best}+r.(X_{i-1}(t)-X_i(t)+e^{bw}.\cos(2\pi w).(X_{best}-X_i(t)) \\ Y_i(Yt+1)=Y_{best}+r.(Y_{i-1}(t)-Y_i(t))+e^{bw}.\sin(2\pi w).(Y_{best}-Y_i(t)) \end{cases} \quad (3)$$

where the number w is chosen at random from 0 to 1.

The motion behavior described here can be extended to d-dimensional space as well. In order to simplify the concept of cyclone foraging, the following mathematical model can be used:

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t)+r.(x_{best}^d(t)-x_i^d(t))+\beta.(x_{best}^d-x_i^d(t)), i=1 \\ x_{best}^d(t)+r.(x_{i-1}^d(t)-x_i^d(t))+\beta.(x_{best}^d(t)-x_i^d(i)), i=2,\dots,N \end{cases} \quad (4)$$

$$\beta = 2e^{r_1(\frac{T-t-1}{T})}.\,sin(2\pi r_1) \quad (5)$$

The weight coefficient is β, the maximum number of iterations is T, and r_1 is the random number between 0 and 1.

Each individual searches randomly for food using the reference position as a guide. Hence, cyclone foraging remains the most efficient method of exploitation in this region. Using this behavior improves exploration as well. Within the entire search space, each individual assigns a random position as his or her reference position, thereby searching far from the best position currently found. In this mechanism, MRFO is primarily concerned with exploration and is able to perform an extensive global search. This mechanism can be described mathematically as follows:

$$x_{rand}^d = Lb^d + r.(Ub^d - Lb^d) \quad (6)$$

$$X_i^d(t+1) = \begin{cases} x_{rand}^d(t)+r.(x_{rand}^d(t)-x_i^d(t))+\beta.(x_{rand}^d-x_i^d(t)), i=1 \\ x_{rand}^d(t)+r.(x_{i-1}^d(t)-x_i^d(t))+\beta.(x_{rand}^d(t)-x_i^d(i)), i=2,\dots,N \end{cases} \quad (7)$$

There are Ub^d and Lb^d upper and lower limits of the *d-th* dimension, respectively, for x^d_{rand}, the position generated randomly in the search space.

Somersault foraging: Food's position in this behavior is considered pivotal. As individuals swim around the pivot, somersaulting to a new position, they tend to swim to and from around it. In order to keep their positions as accurate as possible, they update them around the best position they have found thus far. Modeling can be done in the following way:

$$x_d^i(t+1) = x_d^i(t) + S.(r_2.x_{best}^d - r_3.x_i^d(t)), i = 1, \ldots, N \quad (8)$$

Manta rays' somersault range is determined by *S*, the somersault factor. r_2 and r_3 are two random numbers in the range 0 to 1.

As shown in Equation 8, an individual can move to any positions within the somersault range if the current position is located between the symmetrical position and his best position. Reduced distance between an individual's current position and the best position is accompanied by reduced perturbations on the current position. Throughout the search space, each individual approaches the optimal solution gradually. As iterations increase, somersault foraging range becomes adaptively reduced.

2.5. Learning model

The purpose of this section is to discuss the learning models that are used.

Random forest classifier: With an RF, a number of decision trees are used, each of which acquires its position arrangement effect through dissimilar classification. It is also particularly suitable for some minute models as it allows evaluation of sample allocation based on random sampling. According to RF, the basic classification procedure is as follows:

1. Develop an illustration set that contains *X* cases and *Y* characters;
2. A second training set is created based on substitution bootstrapping by sampling the RF *n* times; this results in a subordinate training set;
3. A certain number of characteristics are selected at random from all distinctiveness when this technique chooses non-leaf nodes (internals). By using these criteria, it divides the nodes optimally. The number of characters that are tried at each division is indicated by *mtry*, $mtry \leq M$;
4. Trees expand more when pruned;
5. Trees created with RF are joined. An entity choice is resolute by a mass selection of the trees in the RF, which transmits its entity choice for the most accepted group;
6. Given that set *S* consists of *k* types of attribute principles, and that each type of attribute principle defines one sub node, *Gini(i)*, one can calculate the Gini coordinates of node *i* as follows:

$$Gini(i) = 1 - \sum_{j=1}^{h}[p(i/j)^2] \quad (9)$$

where *h* is the number of categories of node *i* and *p(j/i)* is the comparative frequency of form no *j* on node *i*.

Obviously, there must be no infection if every node is of the same class, indicating that the main data or entropy has been used. Whenever a node is evenly divided between every class, *Gini(i)* should be elevated as it is known that the divided uses even the smallest amount of valuable information.

7. Set S has the following part index:

$$Gini_{split}(S) = \sum_{i=1}^{r}\frac{S_i}{S}Gini(i) \quad (10)$$

where *S* has a split index of $Gini_{split}$, *r* denotes the type of record within set *S*, S_i denotes the number of records on node *i*, and *S* denotes the number of records in set *S* as a whole.

KNN classifier: Test samples and training samples are loaded into databases according to the diameter closest to the preparation case. After categorizing the sample, its part concludes. By capturing the adjacent *k* positions with proclaiming the mainstream, the KNN classifier expands on this suggestion. The choice of *k* values is unique. Choosing the value of *k* is frequently performed during cross-validation in order to reduce the effects of noisy pixels within the training data set. Greater values of *k* reduce the noise in the pixels rate. It is possible to achieve classification using the 1-NN rule using values from numerous subsets of training data in this case by using several methods [20].

3. Review of related works

A feature selection algorithm reduces the feature size while maximizing model generalization [21]. The Hybrid Binary Bat Particle Swarm Optimization Algorithm was proposed by Tawhid and Dsouza [2] for solving feature selection problems. HBBEPSO is a combination of bat algorithm with its echolocation capability for exploring the feature space, and improved particle swarm optimization that converges to the best global solution. A comparison of the HBBEPSO algorithm with the original optimizers and other features selection optimizers is conducted to investigate the general performance of the proposed algorithm. The proposed HBBEPSO algorithm proved to be capable of searching the feature space for the optimal combination of features. Moslehi and Haeri [21] proposed a hybrid wrapper-filter approach based on genetic algorithms. In the proposed method, dubbed smart HGP-FS, Artificial Neural Networks (ANNs) are used as fitness functions. With the combined use of the filter and wrapper methods, they can use the acceleration provided by the filter and the vigor provided by the wrapper to select datasets with effective features. The filter phase eliminates many of the characteristics of the dataset, which reduces computation complexity and search time in the wrapper. There have been several comparisons of the effectiveness and usability of multiple methods, including the proposed hybrid algorithm, two pure wrapper algorithms, two pure filter procedures, and two traditional wrapper methods for selecting feature sets. Based on real-world datasets, the algorithm was found to be efficient.

Based on Harris Hawks optimization, Abdelbasset et al. [4] presented simulated annealing for feature selection. HHOBSA is a hybrid version of the Harris Hawks Optimization algorithm that uses simulating annealing and bitwise operations to solve the FS problem for classification purposes. By combining two bitwise operations (AND and OR), the most informative features can be randomly transferred from the best solution to other populations. The Simulation Annealing (SA) process improves HHOBSA's performance and finds local optima. Standard wrapper methods for evaluating the new solutions include K-nearest

neighbors with Euclidean distance metrics. HHOBSA is analyzed using 24 standard datasets and 19 artificial datasets with dimensions ranging from tens to thousands, and its performance is compared with other state-of-the-art algorithms. Data dimensions, noise ratios, and sample sizes are used as parameters to study how the FS process is affected. Performance of the proposed algorithm could not be matched by other algorithms.

Using a Harris Hawks optimization algorithm with Simulated Annealing, Elgamal et al. [22] proposed an improved feature selection algorithm based on Harris Hawks. This paper proposes a metaheuristic optimizer based on the chaotic Harris Hawks algorithm (CHHO). To improve the standard HHO algorithm, two main modifications are suggested. To enhance the diversity of the population in the search space, chaotic maps should be applied to the initialization phase of HHO. As a second improvement, the current best solution is improved by using Simulated Annealing (SA). Compared to the standard HHO algorithm and other optimization algorithms, CHHO has demonstrated superior performance on the majority of medical datasets. According to Ding et al. [23], feature selection could be achieved by combining genetic algorithms with competitive swarm optimization techniques. It has been proven to be effective at designing high-dimensional feature selection algorithms with a competitive swarm optimizer based on particle swarm optimization algorithms. Although it has the advantage of being easy to compute, it is also prone to premature execution due to its high computation time costs. The crossover and mutation operators are used in this paper to improve generation speed and prevent premature population growth. As a result of the new algorithm, the competitive swarm optimization algorithm is more efficient and avoids the local optimum problem, which was observed when testing it on the UC Irvine Machine Learning Repository. The problem-specific genetic algorithm proposed by Zhou et al. [24] is a non-dominated sorting algorithm for supervised feature selection. In this paper, the authors propose a problem-specific Non-Dominated Sorting Genetic Algorithm (PS-NSGA) in order to minimize three FS objectives. The PS-NSGA uses a dominance operator that favors accuracy, increasing the chances that individuals with higher classification accuracy will survive. Quick bit mutations make bit string mutations faster and overcome the limitations associated with traditional bit strings. The combination operator and mutation-retry operators are also designed to improve our algorithm's convergent speed. A strategy for selecting the most appropriate feature subset is developed from the obtained Pareto solutions. In a comparison between the proposed algorithm and some existing evolution-based FS algorithms, experimental results indicated that the proposed algorithm obtained a smaller subset of features while achieving competitive classification accuracy.

Six related evolutionary algorithms for feature selection are compared in Table I. Most of these methods do not use high dimensional datasets and do not consider correlations between features. They also had high complexity. It was attempted to solve many of these problems in this article.

4. The proposed method

In this section, a framework for selecting the most relevant features from a dataset is presented, based on the concept of meta-heuristics. This model aims to find the optimal combination of several recent FS solutions that have proven successful. A general framework for the method can be seen in Figure 4, which is divided into two phases (i.e., filtering and wrapping).

Table 1. Related works on feature selection

Ref.	Year	Algorithm(s)	Compared Methods	Objective Function(s)	Learning algorithm	Dataset used	Disadvantage(s)
Tawhid and Dsouza (2018) [2]	2018	HBBEPSO	BBO PSO	Accuracy, Number of selected features	KNN	20	-Small scale problem
Moslehi and Haeri (2020) [21]	2020	HGP-FS	GA PSO	Accuracy, Number of selected features	ANN	6	-The number of datasets is low
Elgamal et al. (2020) [22]	2020	CHHO	HHO SA	Accuracy, Number of selected features	KNN -	14	-Didn't use large datasets
Ding et al. (2020) [23]	2020	HBCSO	PSO GA	Accuracy, Number of selected features	KNN	5	-The number of datasets is low, High running time
Zhou et al. (2021) [24]	2021	PS-NSGA	GA	Accuracy, Number of selected features	KNN NB	15	-High complexity
Abdel-Basset et al. (2021) [4]	2021	HHOBSA	HHO SA	Accuracy, Number of selected features	KNN	24	-Use of low-dimensional data, High complexity

4.1. Filter phase

Material data science commonly uses numerical features in ML models because it is applicable to more models and reflects how key features affect properties intuitively. A data-driven correlation analysis technique can be used to assess the correlation between numerical features and the materials domain knowledge. Correlation coefficients between two features can be calculated quickly using data-driven correlation analysis techniques. In order for two features to be classified as highly correlated, their correlation coefficient must exceed a predetermined threshold. In this work, Spearman correlation coefficient was used to measure the correlation between two features. According to Equation 11, there are different trigger conditions for the different correlation analysis methods. The SCC is capable of measuring correlations between two features (both linear and nonlinear). SCC is proportional to how strongly two features are correlated. The higher the SCC value, the stronger the correlation. Additionally, the absolute value of SCC between f_i and f_j should exceed the correlation threshold k_1 when their correlation is high [25].

$$corr(f_i, f_j) = |SCC(f_i, f_j)|, n \geq k_1 \quad (11)$$

where f_i and f_j are two distinct features in dataset and k_1 represents predefined threshold.

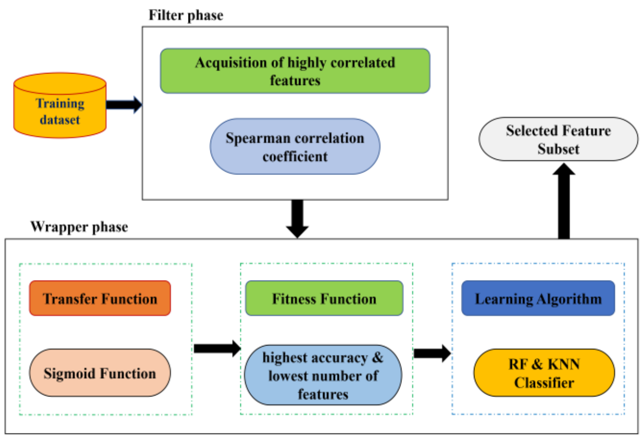


Figure 4. General framework of the proposed method

4.2. Wrapper phase

This section discusses the wrapper phase.

1) Binary Manta Ray Foraging Optimization (BMRF)

Assume that the original feature set $F = [f_1, f2, ..., f_D]$ has dimension D, and that the class label $C = [c_1, ..., c_L]$ has dimension l. FS method finds a subset $S = \{s_1, ..., s_m\}$, where $m < D$, $S \subset F$ and the classification error rate for S is significantly lower than for any other subset of the same size or for any reasonable subset of S.

The solution of FS is restricted to binary values between 0 and 1. In this case, a binary vector represents a solution, in which 1 indicates the corresponding feature has been selected and 0 indicates that it has not been selected. As in the original dataset, this vector is the same size as the number of features. For continuous optimization problems with real values as solutions, MRFO is proposed. A transfer function is used to map the continuous search space of the standard MRFO to a binary search space. According to Equation 12, we used Sigmoid transfer function. An example of a transfer function for converting a continuous search space to a binary search space can be seen in Figure 5.

$$T(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

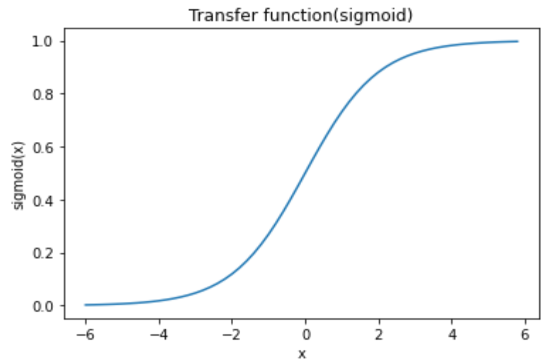


Figure 5. Transfer function for converting continuous search space to binary

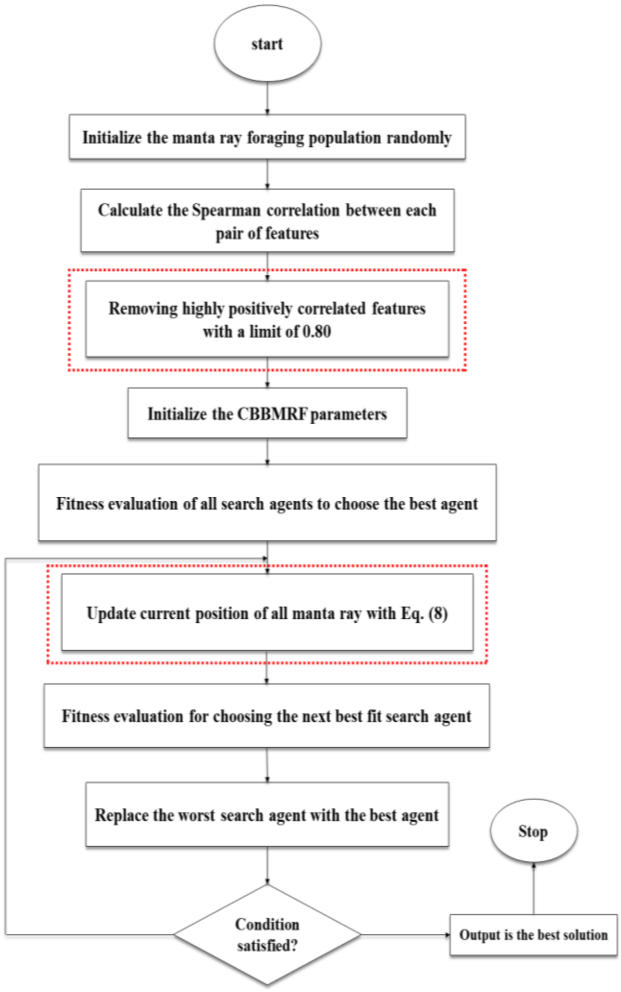


Figure 6. Flowchart of the proposed method

Now, using the probability values generated by Equation 12, the Manta Ray's current position will be updated based on Equation 8.

2) Correlation Based Binary Manta Ray Foraging (CBBMRF)

Figure 6 shows the flow chart of the proposed method (CBBMRF). By considering the correlation threshold at 0.80 [22], we improved the performance of the BMRF method. Based on the evaluation results of both proposed methods, CBBMRF clearly shows the superiority and improved performance over BMRF.

3) Fitness function

In general, FS involves two objectives: Maximization (maximizing classification accuracy) and minimization (selecting the fewest number of features).

There is a contradiction between these two objectives. The classification error rate was taken into account as a means of eliminating this contradiction. A single objective problem is created by combining these two objectives using Equation 13.

$$Fitness = \omega\gamma(S) + (1-\omega)\frac{|S|}{D} \quad (13)$$

Considering the subset of features selected as S, $|S|$ represents the number of features selected, $\gamma(S)$ is classification error rate of S, the dataset's original dimension is D, weights (ω) are represented by the values 0 and 1.

5. Experimental results and discussion

5.1. Experiment setup

Using the proposed feature selection method, we selected a feature subset in order to assess classification accuracy using KNN and RF classifiers [20].

KNN classifier has K= 5 and RF classifier has n-estimators=300 as recommended in [26]. A training dataset consists of 80% of the instances, while a testing dataset consists of the remaining 20%. A subset of features for the selected feature set has been selected using FS methods applied to the train data. Only those features are selected from test data, and then KNN classifiers and RF classifiers are used to measure classification accuracy. Graphs are plotted using Matplotlib, and Python3 is used to implement the proposed method.

A total of 10 UCI datasets were considered to assess the performance of BMRF and CBBMRF. There is a wide variety of backgrounds represented in the datasets. A description of each of these datasets can be found in Table 2. According to Table 2, there are nine bi-class datasets and one multi-class dataset. There is a great deal of diversity in both the number of attributes (features) and the number of instances in these datasets. As a result of these variances, the proposed methods are able to demonstrate their robustness.

Table 2. Description of the datasets used in this work

Sl. No.	Dataset	No. of Attributes	No. of Samples	No. of Classes	Dataset Domain
1	Algerian forest fires	14	244	2	Biology
2	Breast cancer	11	699	2	Biology
3	Scadi	206	70	7	Life
4	DataR2	10	116	2	Biology
5	Wholesale customers	8	440	2	Business
6	Pd speech features	755	756	2	Biology
7	Sonar	61	208	2	Biology
8	Prison	34	48	2	Life
9	Sobar72	20	72	2	Life
10	Visnights	21	76	2	Life

Table 3. Description of the datasets used in the present work with correlation.

Sl. No.	Dataset	No. of Attributes	No. of Samples	No. of Classes	Dataset Domain
1	Algerian forest fires	11	244	2	Biology
2	Breast cancer	10	699	2	Biology
3	Scadi	150	70	7	Life
4	DataR2	9	116	2	Biology
5	Wholesale customers	7	440	2	Business
6	Pd speech features	444	756	2	Biology
7	Sonar	39	208	2	Biology
8	Prison	19	48	2	Life
9	Sobar72	19	72	2	Life
10	Visnights	17	76	2	Life

5.2. Simulation results

1) Impact of correlation

Table 3 shows how the feature dimension changes after correlation is applied. With higher feature dimensions, we see an increase in the probability that features will be correlated, and as a result, more features will be discarded. Conversely, when faced with smaller feature dimensions, as shown in Table 3, there is a greater probability that fewer features will be targeted. The correlation coefficient will then be effective depending on the data feature dimensions.

2) Convergence rate of best fitness value

The size of the population and the number of maximum iterations are always very important parameters for any multi-agent evolutionary algorithm. Iterations provide step-by-step evolution of agents based on the experiences of other agents, whereas population size determines how an agent learns from the experiences of other agents. The fitness function with KNN classifier is shown in Figure 7, and with RF classifier is shown in Figure 8.

As shown in Figure 7, CBBMRF achieves the best fitness value sooner than BMRF in 6 cases (60%). A balanced fitness value can be reached with both methods proposed (BMRF and CBBMRF) in Figure 8. As a result, CBBMRF contributes to the improvement of BMRF in order to reach the best fitness level.

3) Comparison of BMRF and CBBMRF

Results from two classifiers (KNN&RF) are presented here for BMRF and CBBMRF methods.

According to Table 4, BMRF achieved 96% accuracy in 5 cases (50%), while CBBMRF reached 96% accuracy in 7 cases (70%). The same results are shown in Table 5, in which BMRF reached 96% accuracy in 5 cases (50%) while CBBMRF reached 96% accuracy in 7 cases (50%). Table 4 shows that BMRF is 100% accurate in three out of four cases (30%), whereas CBBMRF is 100% accurate in four out of five cases (40%). A total of 4 (40% of) cases were reached by both methods in Table 5. CBBMRF significantly reduces the selected features compared to BMRF in both classifiers, which demonstrates the benefits of adding the filter phase.

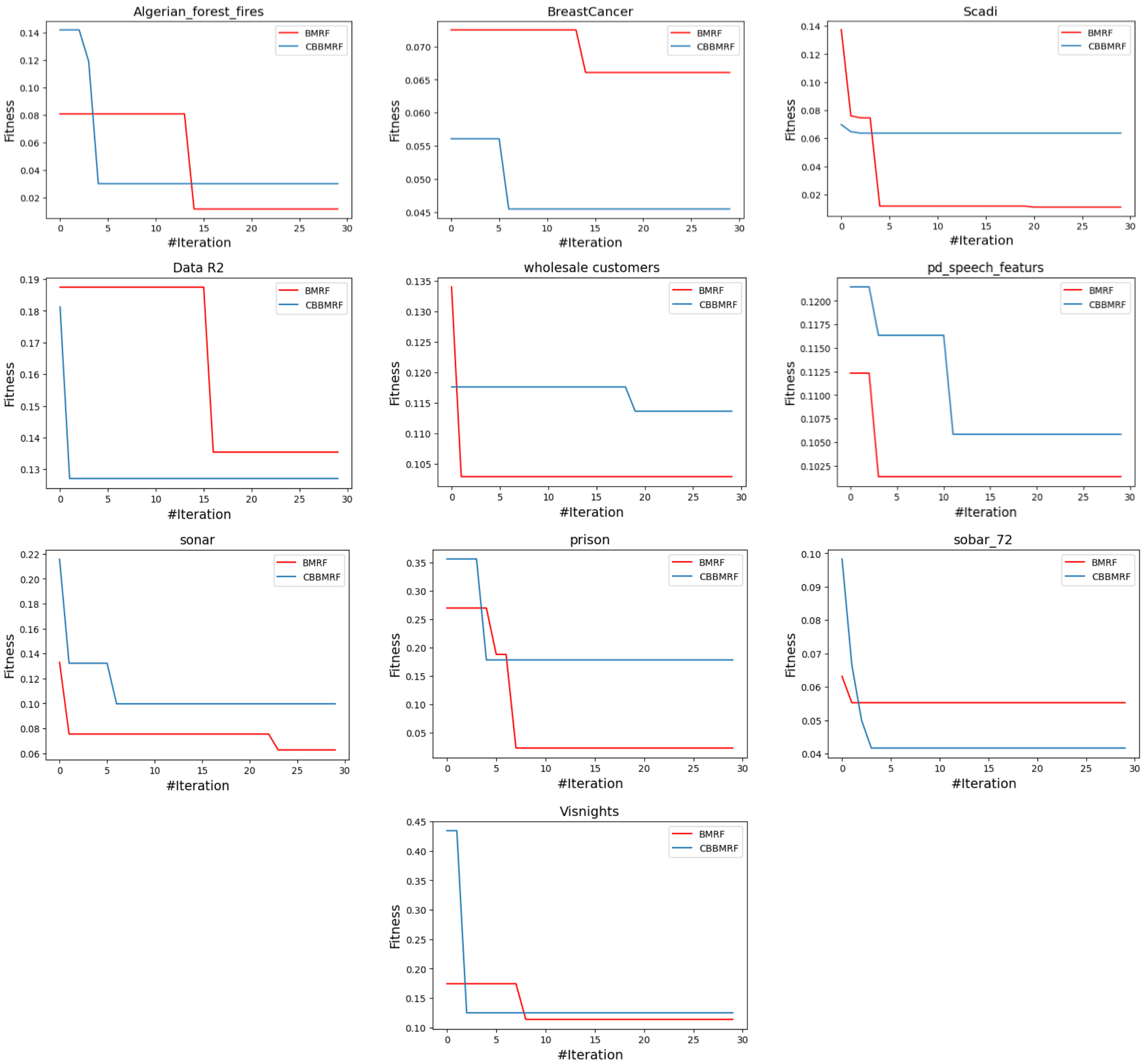


Figure 7. Fitness values using BMRF and CBBMRF (KNN classifier) for ten UCI datasets

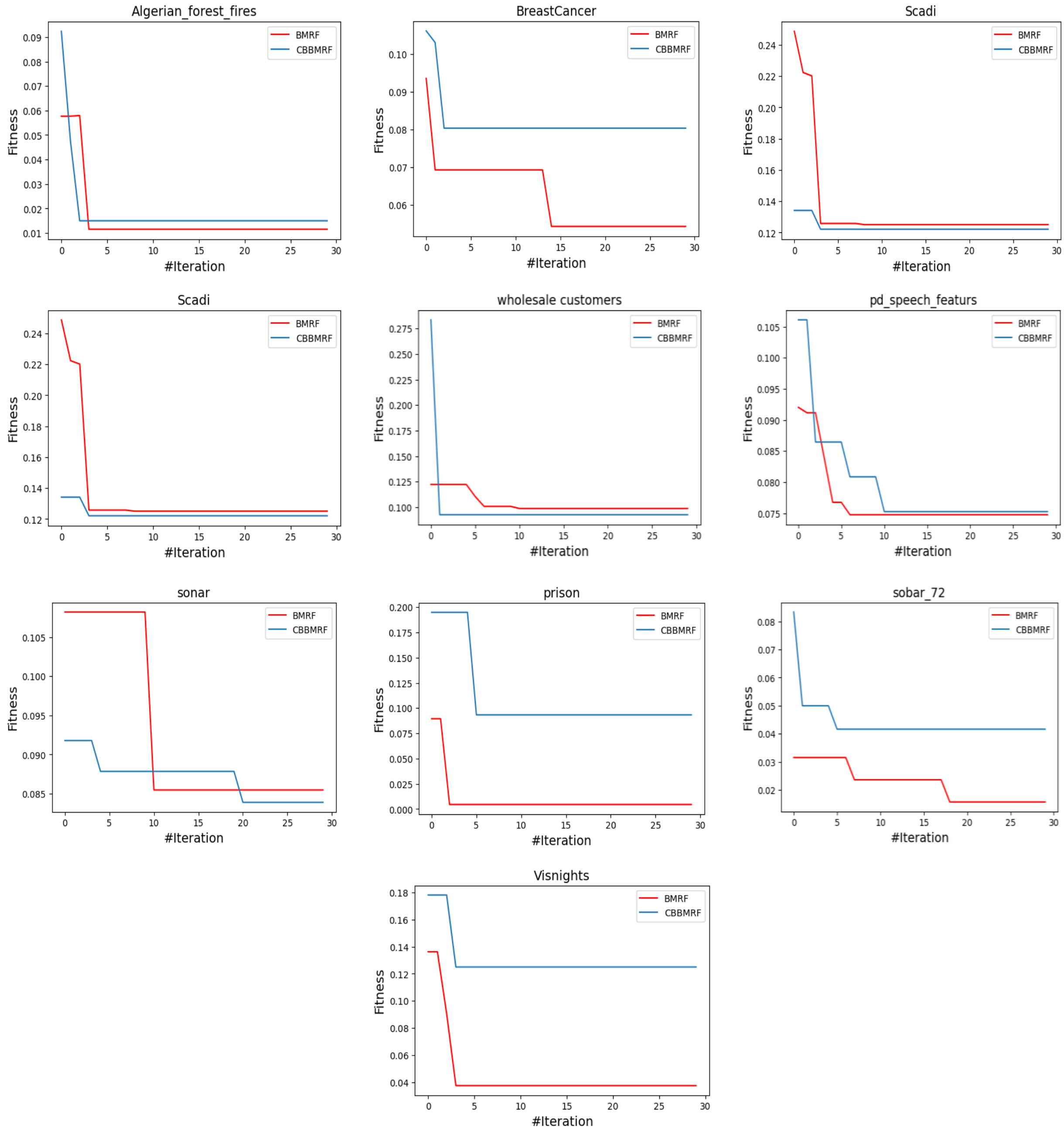


Figure 8. Fitness values using BMRF and CBBMRF (RF classifier) for ten UCI datasets

Table 4. Performance of BMRF and CBBMRF in terms of classification accuracy and selected no. of features (KNN classifier)

Sl. No.	Dataset	BMRF		CBBMRF	
		Accuracy	**#Features**	**Accuracy**	**#Features**
1	Algerian forest fires	100	1	100	1
2	Breast cancer	99.28	5	99.28	2
3	Scadi	92.85	2	100	8
4	DataR2	95.83	5	95.83	4
5	Wholesale customers	95.45	2	98.86	1
6	Pd speech features	93.42	161	94.73	71
7	Sonar	97.61	19	97.61	25
8	Prison	100	1	100	2
9	Sobar72	100	2	100	1
10	Visnights	93.75	2	93.75	3

Table 5. Performance of BMRF and CBBMRF in terms of classification accuracy and selected no. of features (RF classifier)

Sl. No.	Dataset	BMRF		CBBMRF	
		Accuracy	#Features	Accuracy	#Features
1	Algerian forest fires	100	1	100	1
2	Breast cancer	98.57	4	97.95	4
3	Scadi	100	69	100	18
4	DataR2	95.83	4	96.59	1
5	Wholesale customers	100	4	100	2
6	Pd speech features	92.10	206	98.02	1
7	Sonar	95.23	9	95.23	4
8	Prison	90	2	80	1
9	Sobar72	100	1	100	1
10	Visnights	93.75	9	93.75	1

Table 6. Setting parameters for state-of-the-art methods

Algorithm	Parameters
BGA	Pop-Size = 10 Max-iter = 20 Crossover-prob = 0.6 Muprob-min = 0.01 Muprob-max = 0.3
BPSO	Pop-Size = 20 Max-iter = 30 C1, C2 = 2 WMAX = 0.9 WMIN = 0.4
BGR	golden = (1 + 5 ** 0.5) / 2 pop-Size = 10 max-Iter = 10
BSMO	Pop-Size = 10 Max-Iter = 20
BASO	Pop-size = 10 Max-iter = 30 α =50 β= 0.2

5.3. Discussion

In order to verify the applicability of the proposed methods, we compared them with five state-of-the-art approaches: GA, PSO and three new meta-heuristic approaches, including Golden Ratio (GR), Social Mimic Optimization (SMO) and Atom Search Optimization (ASO). Table 6 describes the control parameters of these methods.

1) BMRF evaluation

In feature selection, the accuracy of classification is one of the main criteria for evaluating a method's performance and superiority over other methods. BMRF's classification accuracy is shown in Tables 7 and 8 with KNN and RF classifiers. In all tables, bold numbers indicate the best performance. For each dataset, methods are ranked according to their classification accuracy. Considering the three unique manta ray strategies (i.e., chain foraging, storm search, and somersault foraging), BMRF using the KNN classifier achieved the best accuracy in six cases (60%) including "Algerian forest fires", "Breast cancer", "Pd speech features", "Sonar", "Prison", and "Sobar72". As a result, BMRF outperforms four other methods such as BPSO (the best performance in four datasets or 40%), BGA (the best performance in two datasets or 20%), BSMO (the best performance in five datasets or 50%), and BASO (the best performance in two datasets or 20%).

The BGR optimization technique has unique features; for example, a vector and its direction are used for finding the best solution. In order to determine fitness, the mean of the population should first be calculated, and then a criterion should be calculated. Calculating fitness results in determining the best and worst fitness. In the following step, a random sampling of individuals is evaluated to see which moves have the greatest and least impact on their lives. Therefore, using the same number of datasets (6 datasets), it achieved the highest accuracy through the optimization process. BMRF using RF classifier in 9 datasets (90%) performed better than BGR (50%), BPSO (40%), BGA (40%), BSMO (40%), and BASO (10%) due to the implementation of three unique manta ray strategies. As a result of the suggested method, BMRF performs optimally in comparison with other methods.

Table 7. The comparison of classification accuracy obtained by proposed method (BMRF) with that of other methods for 10 UCI datasets

(KNN classifier)

Dataset	BMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	1	0.9795	1	1
Breast cancer	0.9928	0.9928	0.9857	0.9928	0.9857	0.9885
Scadi	0.9285	0.9285	1	0.9285	0.8571	0.7777
DataR2	0.9583	0.9583	0.9583	0.8750	1	0.9310
Wholesale customers	0.9545	0.9886	0.9659	0.9431	0.9659	0.9636
Pd speech features	0.9342	0.9342	0.9276	0.9210	0.9144	0.8994
Sonar	0.9761	0.9761	0.9761	0.9761	0.9761	0.9230
Prison	1	0.9000	0.9000	0.8000	0.8000	0.8947
Sobar72	1	1	1	0.9333	1	1
Visnights	0.9375	0.9375	0.9375	0.7500	1	0.8994

Table 8. The comparison of classification accuracy obtained by proposed method (BMRF) with that of other methods for 10 UCI datasets (RF classifier)

Dataset	BMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	1	1	1	1
Breast cancer	0.9857	0.9857	0.9785	0.9857	0.9857	0.9771
Scadi	1	0.9285	0.8571	1	0.9285	0.9444
DataR2	0.9583	0.9583	0.8750	0.8333	0.9166	0.8275
Wholesale customers	1	0.9545	0.9886	0.9545	0.9659	0.9545
Pd speech features	0.9210	0.9210	0.9078	0.8815	0.9276	0.9100
Sonar	0.9523	0.9761	0.9523	1	0.9761	0.9423
Prison	0.9000	0.8000	0.9000	0.7000	0.9000	0.6666
Sobar72	1	1	1	0.9333	1	0.8888
Visnights	0.9375	0.7500	0.8125	0.5625	0.8125	0.8947

In addition to selecting the best set of features, one must also determine how many features are selected. Among the different methods, the method that selects the best and smallest feature set will perform best. BMRF and other methods using KNN classifiers and RF classifiers each select a different number of features, as shown in Tables 9 and 10. Table 9 shows that KNN classifier selects the minimum feature in 50% of the datasets (Algerian forest fires, Scadi, Wholesale customers, Prison, and Sobar72) using the proposed method (BMRF). BSMO selected the minimum feature in 1 dataset (10%), BPSO selected it in 3 datasets (30%), BASO selected it in 3 datasets (30%), BGR selected the minimum feature in 2 datasets (20%) and BGA in 1 dataset (10%).

Furthermore, Table 10 (RF classifier) shows that in five out of ten datasets (50%), the proposed method selected a minimum feature level, whereas it did poorly in other five datasets and selected more features. While BMRF performed better than four methods, BGR, BGA, BSMO, and BASO, only BPSO outperformed others, as it selected a minimum feature level in seven datasets. According to these interpretations, the proposed BMRF is more efficient than other methods in selecting the minimum feature.

Based on Figure 9, the proposed method (BMRF) achieved the most accuracy and selected the lowest average features over the 10 UCI datasets (KNN and RF) compared to the five state-of-the-art methods.

Table 9. The comparison of number of selected features obtained by proposed method (BMRF) with that of other methods for 10 UCI datasets (KNN classifier)

Dataset	BMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	2	3	2	3	4
Breast cancer	5	2	2	3	3	47
Scadi	2	59	15	19	42	5
DataR2	5	4	5	5	7	4
Wholesale customers	2	3	3	4	4	17
Pd speech features	161	227	151	278	162	255
Sonar	17	22	24	26	11	3
Prison	1	9	3	1	1	4
Sobar72	2	3	2	3	5	9
Visnights	2	7	5	4	11	1

Table 10. The comparison of number of selected features obtained by proposed method (BMRF) with that other methods for 10 UCI datasets
(RF classifier)

Dataset	BMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	1	2	1	3
Breast cancer	4	2	2	2	3	3
Scadi	69	63	3	95	36	47
DataR2	4	3	3	4	5	5
Wholesale customers	4	2	1	4	3	3
Pd speech features	70	281	119	80	256	171
Sonar	9	26	27	23	45	19
Prison	2	6	2	4	11	3
Sobar72	1	3	3	6	3	8
Visnights	9	9	1	1	4	2

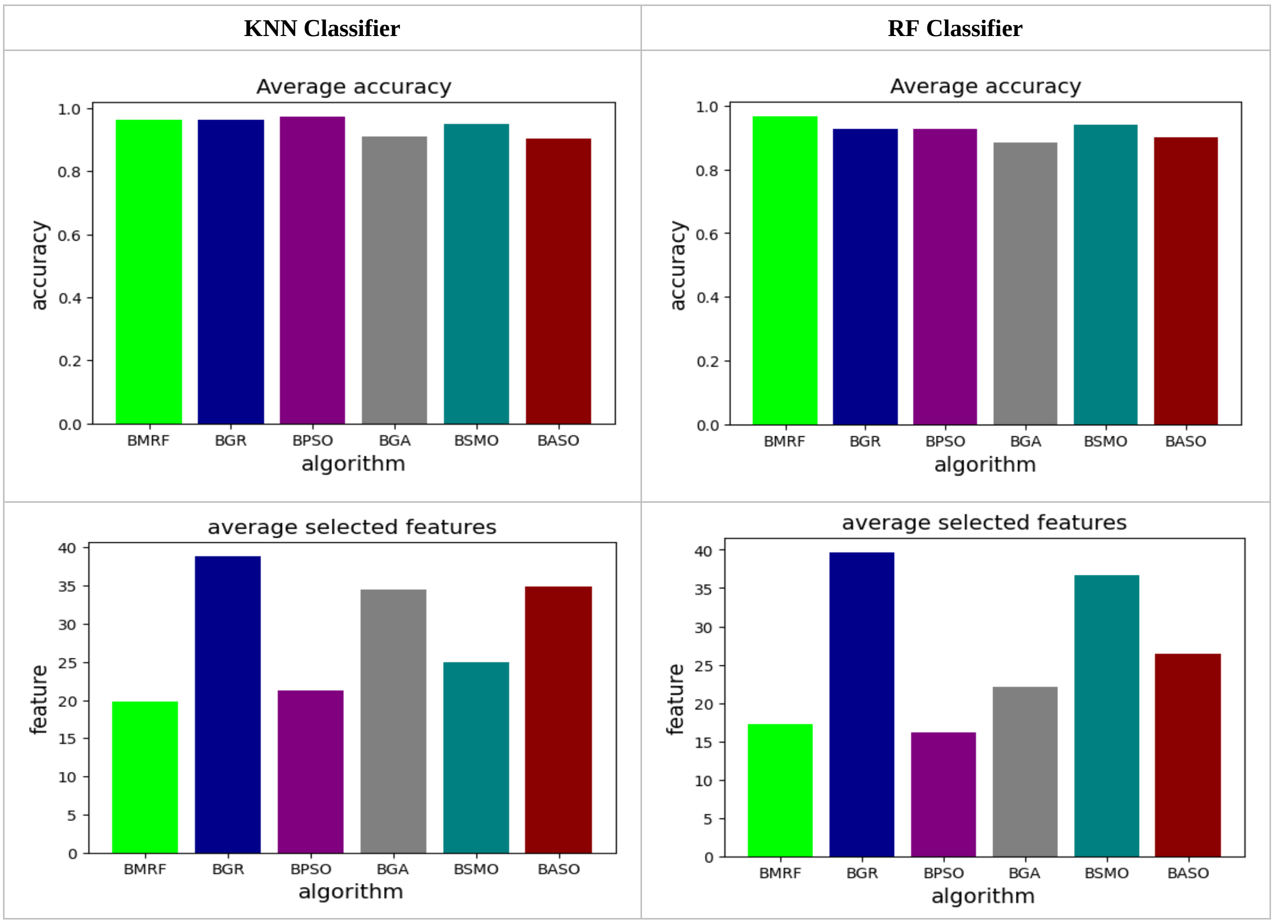


Figure 9. Average accuracy achieved and average number of features selected by the proposed method (BMRF) and 5 state-of-the-art methods over the utilized 10 UCI datasets (KNN and RF classifier)

2) CBBMRF evaluation

According to Tables 11 and 12, CBBMRF offers high classification accuracy when using KNNs and RF classifiers. Based on the classification accuracy of the corresponding dataset, each method is ranked. Average rankings are calculated based on ten UCI datasets. The methods are ranked based on their average rank. A high level of performance can be seen in Table 11, where CBBMRF performs best in each 10 cases (100%). After that, BGR has had the best performance in 8 datasets due to its unique ability to find the optimal solution. According to Table 12, CBBMRF performed best in 7 cases (70%) and had a lower classification accuracy in only three datasets, “Breast cancer”, “DataR2”, and “Prison”.

Tables 13 and 14 compare CBBMRF to KNN and RF classifiers. The CBBMRF selected the fewest features (60%) of six datasets. As shown in Table 15, CBBMRF selects the minimum features in seven datasets (70% of the total dataset). Through CBBMRF, where filtering and wrapping are combined, the number of features selected is greatly

reduced. As a result, the proposed method appears to have a higher level of effectiveness. According to both classifiers' results, CBBMRF is more accurate and has a lower number of selected features than BMRF.

Over the used 10 UCI datasets (KNN and RF), Figure 10 shows that the average accuracy achieved by the proposed method (CBBMRF) and the average number of selected features. According to Figure 10, the proposed CBBMRF method shows relatively higher accuracy compared to other methods. Moreover, according to the proposed algorithm, the average number of selected features is less than that of other methods.

Table 11. The comparison of classification accuracy obtained by proposed FS method (CBBMRF) with that of other methods for 10 UCI datasets (KNN classifier)

Dataset	CBBMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	1	0.9795	1	1
Breast cancer	0.9928	0.9928	0.9857	0.9928	0.9857	0.9771
Scadi	1	1	1	1	0.9857	0.8888
DataR2	0.9583	0.9583	0.9166	0.8333	0.9583	0.8620
Wholesale customers	0.9886	0.9659	0.9545	0.9431	0.9659	0.9545
Pd speech features	0.9437	0.9437	0.9408	0.8487	0.8881	0.8994
Sonar	0.9761	0.9761	0.9285	0.9761	0.9761	0.9423
Prison	1	0.9000	1	0.7000	0.9000	0.8421
Sobar72	1	1	0.9333	0.9333	1	1
Visnights	0.9375	0.9375	0.8750	0.8125	0.8750	0.8333

Table 12. The comparison of classification accuracy obtained by proposed FS method (CBBMRF) with that of other methods for 10 UCI datasets (RF classifier)

Dataset	CBBMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	0.9795	0.9795	0.9795	1
Breast cancer	0.9785	0.9785	0.9857	0.9857	0.9714	0.9714
Scadi	1	0.9285	0.9285	1	0.8571	1
DataR2	0.9659	0.9166	0.9166	0.7916	0.8333	1
Wholesale customers	1	0.9204	0.9772	0.9431	0.9545	0.9636
Pd speech features	0.9802	0.9210	0.9078	0.8881	0.9276	0.9206
Sonar	0.9523	0.9047	0.9047	0.9523	0.9047	0.8846
Prison	0.8000	0.6000	0.9166	0.6000	0.6000	0.8000
Sobar72	1	1	1	0.9333	1	1
Visnights	0.8750	0.7500	0.8125	0.5625	0.8750	0.8421

Table 13. The comparison of number of selected features obtained by proposed FS method (CBBMRF) with that of other methods for 10 UCI datasets (KNN classifier)

Dataset	CBBMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	1	1	2	4	2
Breast cancer	2	3	2	4	3	3
Scadi	8	8	17	61	8	69
DataR2	4	5	4	3	6	3
Wholesale customers	4	2	4	4	3	1
Pd speech features	71	299	133	109	119	137
Sonar	25	20	13	9	20	18
Prison	2	5	2	4	1	6
Sobar72	1	2	2	3	3	5
Visnights	2	9	6	4	2	4

Table 14. The comparison of number of selected features obtained by proposed FS method (CBBMRF) with that of other methods for 10 UCI datasets (RF classifier)

Dataset	CBBMRF	BGR	BPSO	BGA	BSMO	BASO
Algerian forest fires	1	3	1	2	2	5
Breast cancer	4	2	2	2	3	2
Scadi	18	49	27	58	15	83
DataR2	1	4	6	4	2	47
Wholesale customers	2	1	3	5	4	2
Pd speech features	1	145	93	117	132	106
Sonar	4	17	16	28	19	17
Prison	1	5	3	1	2	5
Sobar72	1	3	3	2	4	4
Visnights	1	3	4	2	4	4

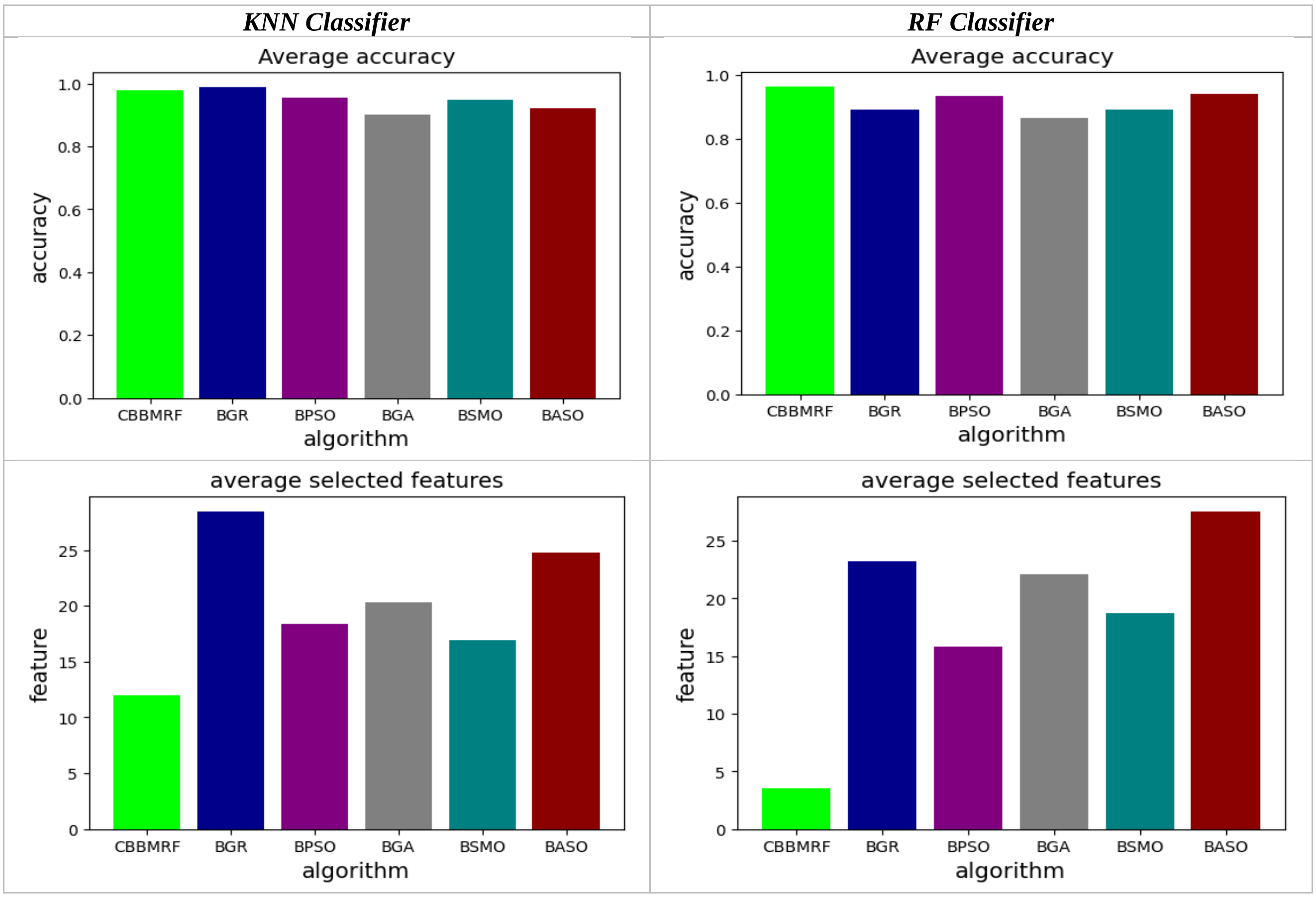


Figure 10. Average accuracy achieved and average number of features selected by the proposed method (CBBMRF) with that of other methods over the used 10 UCI datasets (KNN and RF classifier)

6. Conclusion

Data mining and machine learning have made dimensionality reduction increasingly important in many fields as a result of big data. A binary version of the Manta Ray Foraging Optimizer (MRFO) was developed to simulate three unique manta ray foraging methods: Chain foraging, storm foraging, and somersault foraging. We also combined Spearman's correlation coefficient with our proposed method to reduce costs and calculations, which we called Correlation Based Binary Manta Ray Foraging (CBBMRF). Ten UCI standard datasets were used to evaluate five other algorithms along with BMRF and CBBMRF. To learn classification rules, KNN and RF classifiers were used. It was concluded that the proposed methods BMRF and CBBMRF perform substantially better than the five state-of-the-art metaheuristic FS approaches. Both classification accuracy and feature selection were significantly improved by CBBMRF over BMRF. The robustness and stability of the proposed approach were demonstrated using a range of standard evaluation measures. A more comprehensive objective function will be included in feature selection in the future, as well as further improving the search efficiency of

BMRF for wrapped-based feature selection. To boost the classification performance, multi-objective embedding-based feature selection frameworks and advanced classifiers will also be explored.

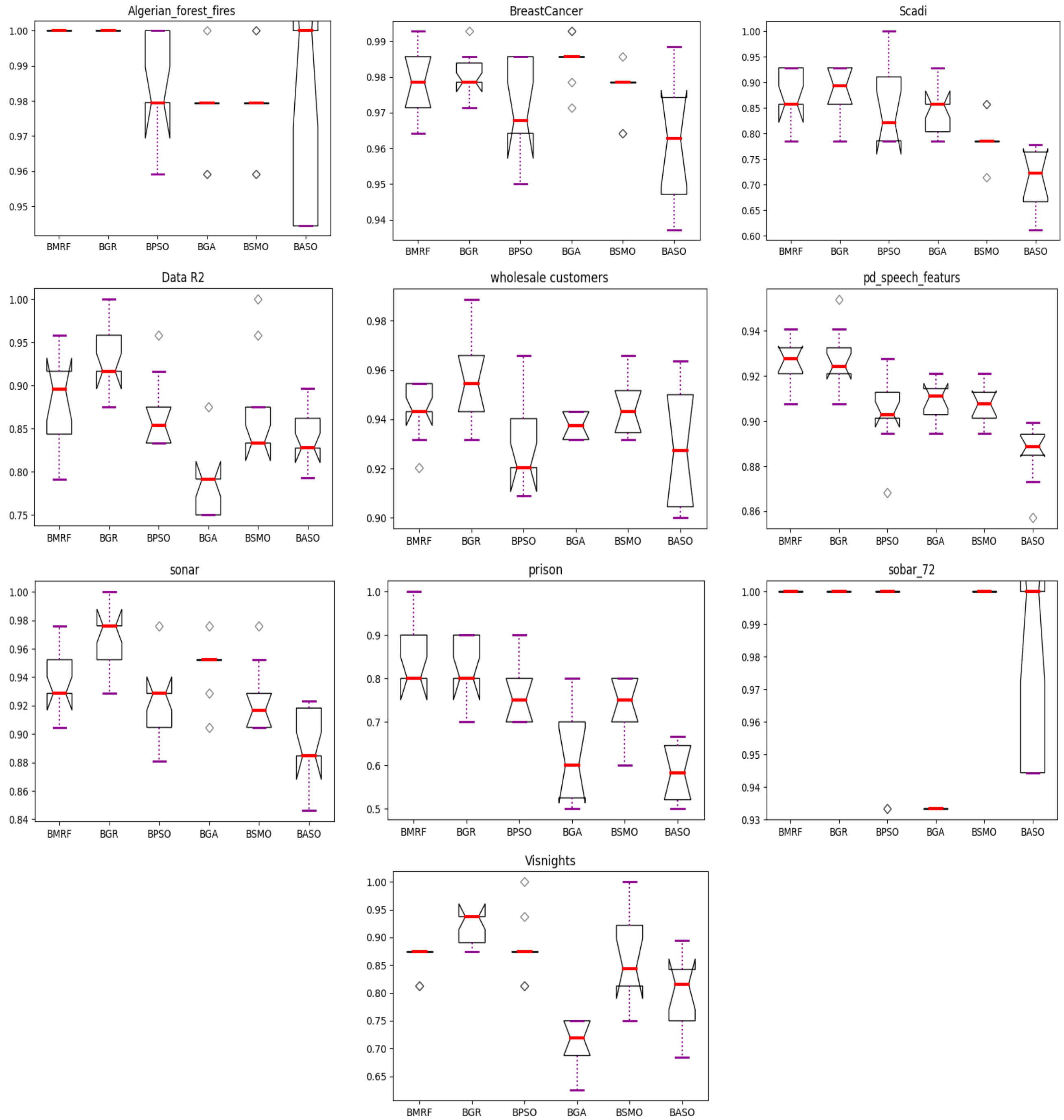


Figure 11. The Boxplot of BMRF and the other method for the 10 UCI datasets (KNN classifier)

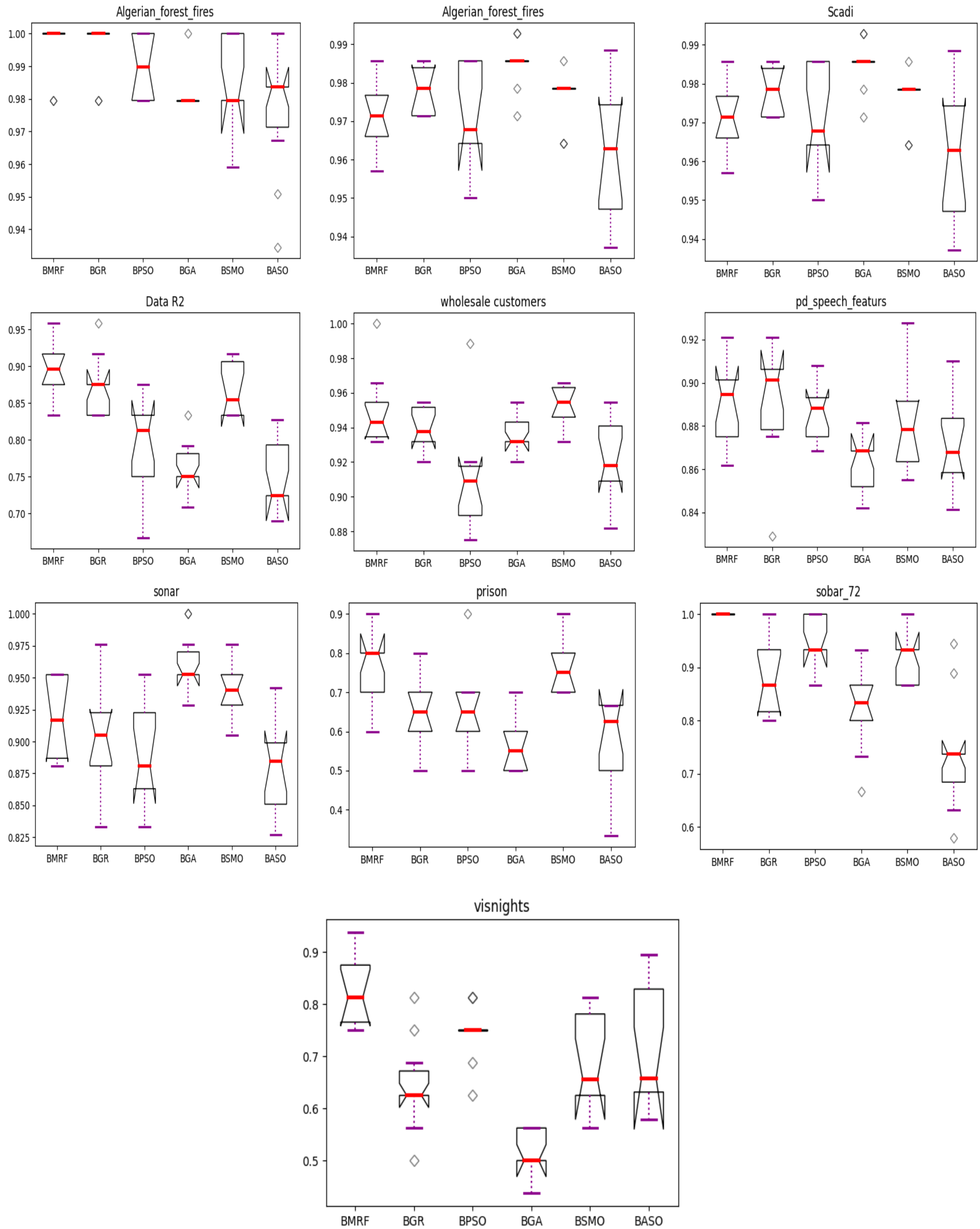


Figure 12. The Boxplot of BMRF and the other method for the 10 UCI datasets (RF classifier)

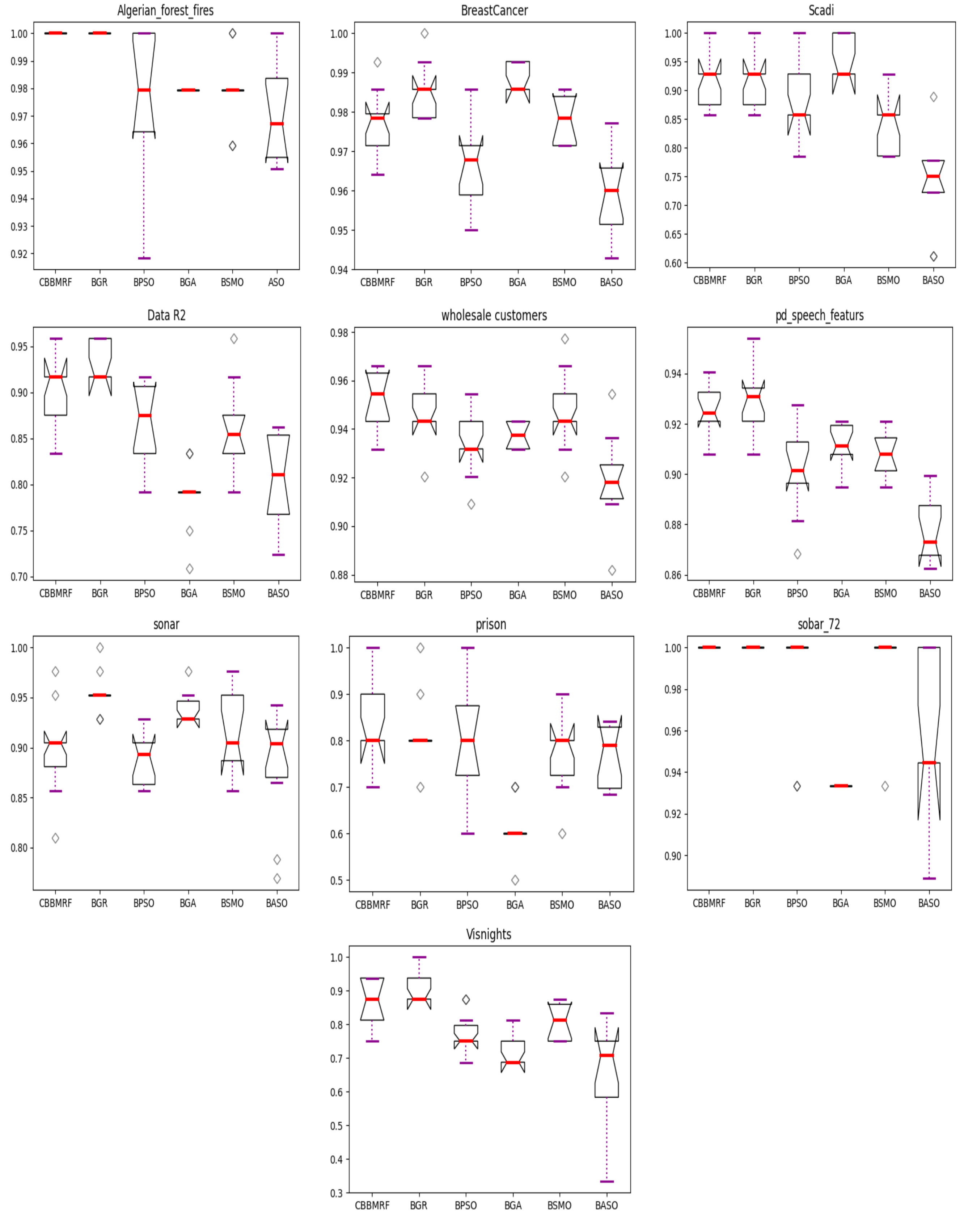


Figure 13. The Boxplot of CBBMRF and the other method for the 10 UCI datasets (KNN classifier)

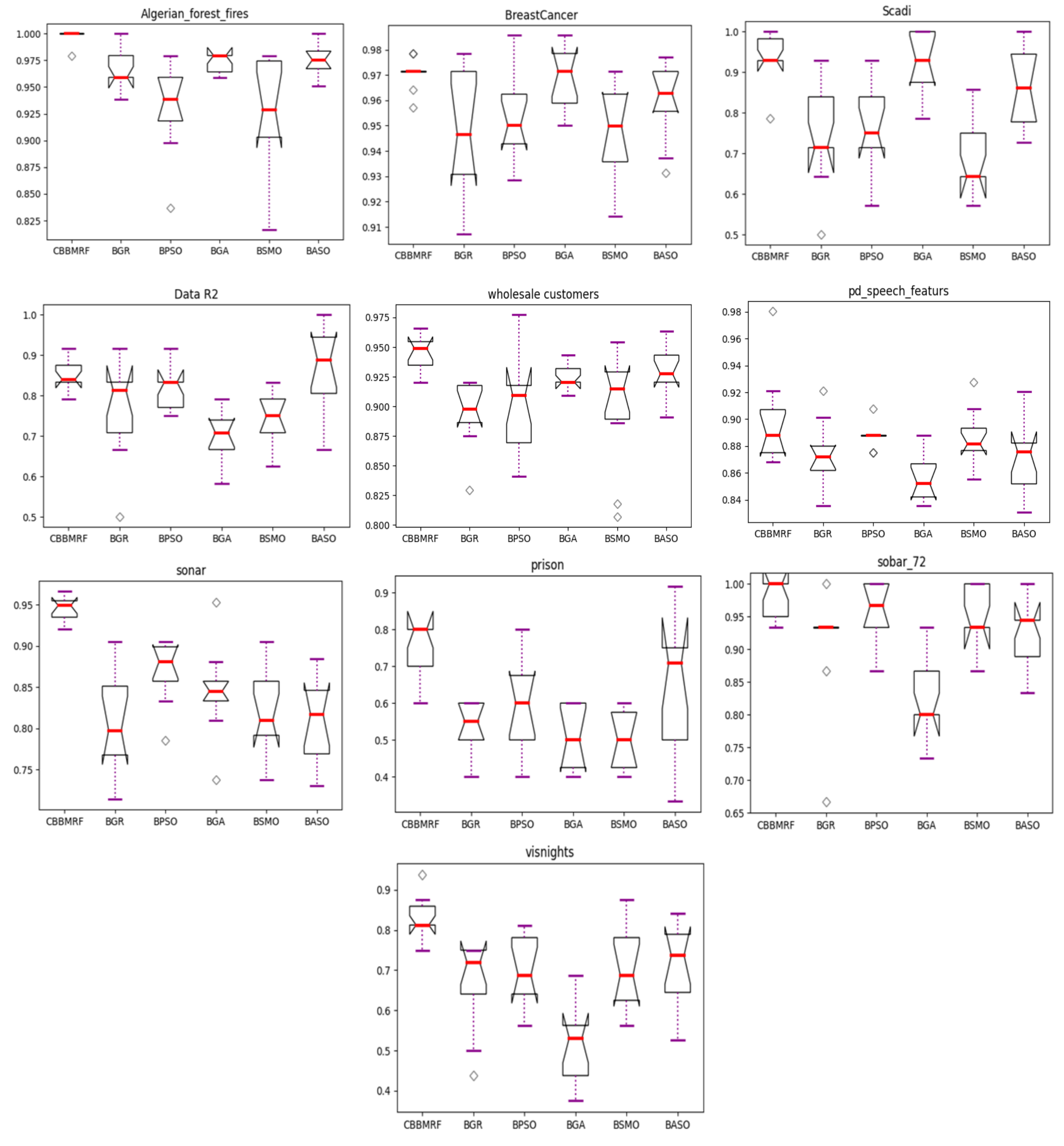


Figure 14. The Boxplot of CBBMRF and the other method for the 10 UCI datasets (RF classifier)

7. Appendix

7.1. BMRF Boxplot

As shown in Figure 11 and Figure 12, all datasets in the proposed BMRF (with KNN and RF classifier) are boxplotted. In data analysis, boxplots are commonly used to display quantitative and qualitative data summaries. A boxplot shows the upper and lower quartiles, the minimum and maximum range values, and the median [27]. BMRF is more stable than most other methods in most datasets. As shown in Figure 11, BMRF method performed better in most datasets, except for "Breast cancer" and "Sonar". In this case, it only performs worse than BGR. Based on Figure 12 the BMRF method performed better than all other methods in 5 datasets (Algerian forest fires, DataR2, Prison, Sobar72, and Visnights).

7.2. CBBMRF Boxplot

Figures 13 and 14 show CBBMRF boxplots with KNN and RF classifiers. CBBMRF is more stable when applied to most datasets. Figure 13 (KNN classifier) shows that the proposed method has a median equal to or higher than other methods in seven datasets (70%). While BMRF performed worse in the previous section, CBBMRF is by far the superior method. As Figure 14 (RF classifier) shows, the proposed method outperforms other methods in 9 datasets (90%).

8. References

[1] A. Adamu, M. Abdullahi, S. B. Junaidu, and I. H. Hassan, "An hybrid particle swarm optimization with crow search algorithm for feature selection", *Machine Learning with Applications,* vol. 6, pp. 100108, 2021.

[2] M. A. Tawhid and K. B. Dsouza, "Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems", *Applied Computing and Informatics*, 2018.

[3] A. M. Anter and M. Ali, "Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems", *Soft Computing,* vol. 24, no. 3, pp. 1565-1584, 2020.

[4] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection", *Artificial Intelligence Review,* vol. 54, no. 1, pp. 593-637, 2021.

[5] L. Abualigah and A. J. Dulaimi, "A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm", *Cluster Computing,* vol. 24, no. 3, pp. 2161-2176, 2021.

[6] O. Tarkhaneh, T. T. Nguyen, and S. Mazaheri, "A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm", *Information Sciences,* vol. 565, pp. 278-305, 2021.

[7] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, "Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection", *IEEE Access,* vol. 8, pp. 83548-83560, 2020.

[8] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications", *Engineering Applications of Artificial Intelligence,* vol. 87, pp. 103300, 2020.

[9] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining", *Swarm and Evolutionary Computation,* vol. 54, pp. 100663, 2020.

[10] D. Jain and V. Singh, "Feature selection and classification systems for chronic disease prediction: A review", *Egyptian Informatics Journal,* vol. 19, no. 3, pp. 179-189, 2018.

[11] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review", *Journal of King Saud University-Computer and Information Sciences,* 2019.

[12] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods", *Engineering Applications of Artificial Intelligence,* vol. 100, pp. 104210, 2021.

[13] M. Lualdi and M. Fasano, "Statistical analysis of proteomics data: a review on feature selection", *Journal of proteomics,* vol. 198, pp. 18-26, 2019.

[14] L. Xie, Z. Li, Y. Zhou, Y. He, and J. Zhu, "Computational diagnostic techniques for electrocardiogram signal analysis", *Sensors,* vol. 20, no. 21, pp. 6318, 2020.

[15] R. A. Kumar, J. V. Franklin, and N. Koppula, "A Comprehensive Survey on Metaheuristic Algorithm for Feature Selection Techniques", *Materials Today: Proceedings*, 2022.

[16] T. Dokeroglu, A. Deniz, and H. E. Kiziloz, "A Comprehensive Survey on Recent Metaheuristics for Feature Selection", Neurocomputing, 2022.

[17] S. Kurman and S. Kisan, "An in-depth and contrasting survey of meta-heuristic approaches with classical feature selection techniques specific to cervical cancer", *Knowledge and Information Systems,* pp. 1-54, 2023.

[18] R. Yadav, I. Sreedevi, and D. Gupta, "Augmentation in performance and security of WSNs for IoT applications using feature selection and classification techniques", *Alexandria Engineering Journal,* vol. 65, pp. 461-473, 2023.

[19] S. M. Ebrahimi and M. J. Hemmati, "Design optimization of the complementary voltage controlled oscillator using a multi-objective gravitational search algorithm", *Evolving Systems,* vol. 14, no. 1, pp. 59-67, 2023.

[20] A. Murugan, S. A. H. Nair, and K. Kumar, "Detection of skin cancer using SVM, random forest and kNN classifiers", *Journal of medical systems,* vol. 43, no. 8, pp. 1-9, 2019.

[21] K. Hussain, N. Neggaz, W. Zhu, and E. H. Houssein, "An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection", *Expert Systems with Applications,* vol. 176, pp. 114778, 2021.

[22] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field", *IEEE Access,* vol. 8, pp. 186638-186652, 2020.

[23] Y. Ding, K. Zhou, and W. Bi, "Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer", *Soft Computing,* vol. 24, no. 15, pp. 11663-11672, 2020.

[24] Y. Zhou, W. Zhang, J. Kang, X. Zhang, and X. Wang, "A problem-specific non-dominated sorting genetic algorithm for supervised feature selection", *Information Sciences,* vol. 547, pp. 841-859, 2021.

[25] Y. Liu, X. Zou, S. Ma, M. Avdeev, and S. Shi, "Feature selection method reducing correlations among features by embedding domain knowledge", *Acta Materialia,* vol. 238, p. 118195, 2022.

[26] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection", *Cognitive Computation,* vol. 12, no. 1, pp. 150-175, 2020.

[27] R. A. Ibrahim, M. Abd Elaziz, A. A. Ewees, M. El-Abd, and S. Lu, "New feature selection paradigm based on hyper-heuristic technique", *Applied Mathematical Modelling,* vol. 98, pp. 14-37, 2021.